

Департамент информатизации Тюменской области
Государственное автономное учреждение дополнительного образования
Тюменской области
«Региональный информационно-образовательный центр»

СОГЛАСОВАНО

Директор
Департамента информатизации
Тюменской области

С.И. Логинов

«16» сентября 2024 г



УТВЕРЖДАЮ

Директор
ГАУ ДО ТО «РИО-Центр»

О.А. Арашкиева

«16» сентября 2024 г



УЧЕБНАЯ ПРОГРАММА

«Основы проектной деятельности в Game Dev»

Трудоемкость программы – 144 академических часа

Форма обучения – очная

Режим занятий – 3 академических часа в день

Начальные навыки: Базовые навыки работы на персональном компьютере

Цель обучения: сформировать у слушателей знания, умения и навыки продвинутого использования языка программирования C# и объектно-ориентированного программирования для разработки приложений для ПК и мобильных устройств, работы в команде, а также знания, умения и навыки, необходимые для создания полноценных игр в игровом движке Unity и их дальнейшей публикации на площадке.

Компетенции на выходе:

Знания:

- углублённый синтаксис языка C# и функционал библиотеки UnityEngine;
- основы геймдизайна и левелдизайна, применяемые в разработке игр;
- принципы проектирования и разработки видеоигр;
- основы проектирования архитектуры проекта;
- базовые паттерны проектирования для GameDev;
- основные знания текстурирования моделей в Substance 3D Painter, Quixel Mixer, Blender;
- основные знания оснастки (рига), скиннинга и анимации 3D моделей;
- основные знания подготовки 3D моделей к экспорту в игровые движки;
- импорт и сборка ассетов, локаций и шаблонных проектов в игровых движках.

Умения:

- использовать библиотеки Dotween, Terrain System, Input System;
- писать независимые от движка Unity системы;
- создавать риг, скин и анимации для 3D моделей;
- работа со ScriptableObjects;
- работа со JSON;
- работа Yandex.Games API;

Приёмы работы:

- при текстурировании;
- при составлении архитектуры проекта;
- при проектировании кодовой базы;
- при работе над средними и крупными играми;

УЧЕБНЫЙ ПЛАН

№ п/п	Наименование тем	Всего часов	в том числе:		Формы контроля
			аудиторные занятия	самостоятельная работа	
1	Командная работа	9	9		Выполнение практических заданий
2	Работа над окружением и наполнением игры	24	24		
3	Продвинутое программирование и моделирование	30	30		
4	Паттерны проектирования, продвинутое моделирование	9	9		
5	Продвинутая работа с движком, продвинутое моделирование	24	24		
6	Проектная деятельность	45	45		
7	Итоговое тестирование	3	3		Выполнение теста
Итого часов:		144	144		

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Наименование тем	Всего часов	в том числе:	
			Теория	Практика
Командная работа (9 часов)				
1.	Введение. Знакомство. Образование команд.	2	1	1
2.	Soft skills. Работа в команде. Командообразование. Team Leader.	2	1	1
3.	Основы проектной деятельности. Основы систем ведения проекта Kanban, Scrum и Agile.	2	1	1
4.	Системы контроля версий. Git.	3	2	1
Работа над окружением и наполнением игры (24 часа)				
5.	Введение в основы геймдизайна. Различные игровые механики на примерах из игр.	3	2	1

6.	Основы сценаристики и нарративного дизайна.	3	1	2
7.	Основы левелдизайна. Создание запоминающейся локации.	3	1	2
8.	Путь игрока. Приёмы и паттерны. Сторителлинг через окружение.	3	2	1
9.	Прототипирование уровней. ProBuilder.	3	1	2
10	Terrain. Работа с ландшафтом.	6	3	3
11	Terrain System. Инструменты. Карты высот.	3	2	1
<i>Продвинутое программирование и моделирование (30 часов)</i>				
12	Основы ООП. 4 базовых принципа ООП. Принцип 1. Абстракция. Художественные стили моделей. Стилизация и реализм.	3	2	1
13	Принцип 2. Инкапсуляция. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.	1	1	0
14	Свойства в С#. Инкапсуляция через свойства. Модификаторы get и set. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.	2	1	1
15	Принцип 3. Наследование. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.	1	1	0
16	Абстрактные классы. Абстрактные и виртуальные методы. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.	2	1	1
17	Интерфейсы в С#. Абстракция vs Наследование. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.	3	2	1
18	Обычные классы vs MonoBehaviour. Сравнение и примеры использования. Способы текстурирования. Детальное,	3	2	1

	процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.			
19	Принцип 4. Полиморфизм. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.	3	1	2
20	Принципы для разработки: KISS, DRY, YAGNI, BDUF, SOLID, APO. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.	3	2	1
21	События в C#. Реализация взаимодействия между скриптами. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.	3	1	2
22	Работа с коллекциями. Массивы vs Списки. Дополнительные структуры хранения данных. Создание UDIM развёртки.	3	2	1
23	Работа с файлами в C#. Чтение и запись файлов. Работа с файлами в Unity. Риггинг, скиннинг и анимация.	2	1	1
24	Scriptable Objects. Применение в архитектуре игры. Риггинг, скиннинг и анимация.	1	0	1
Паттерны проектирования, продвинутое моделирование (9 часов)				
25	Паттерны проектирования. Паттерны «Абстрактная фабрика» и «Фабричный метод». Риггинг, скиннинг и анимация.	3	2	1
26	Паттерн «Машина состояний». 3D сканирование. Оптимизация сканов и моделей для игр.	3	2	1
27	Паттерн «Наблюдатель». Постановка света, камеры и окружения.	1	0	1
28	Паттерн «Медиатор». Постановка света, камеры и окружения.	2	1	1
Продвинутая работа с движком, продвинутое моделирование (24 часа)				
29	Архитектура проекта. Иерархия и структура файловой системы. Постановка света, камеры и окружения.	3	2	1

30	Создание пользовательских пакетов и инструментов для движка. Создание рендера, анимированного превью и тизера.	3	2	1
31	Input System. Обработка ввода для разных устройств. Создание рендера, анимированного превью и тизера.	3	2	1
32	Dotween. Работа с анимациями. Создание рендера, анимированного превью и тизера.	3	2	1
33	Шейдеры. Shader Graph. Создание кастомных шейдеров. Работа с моделями в движке.	6	3	3
34	Light Probes. Углублённая работа с освещением. Создание и оформление портфолио для 3D художника.	3	2	1
35	Использование LOD. Настройка детализации моделей в проекте. Площадки для публикации. Создание страницы и выгрузка работ.	3	1	2
Проектная деятельность (45 часов)				
36	Разработка основных геймплейных механик и скелета игры.	15	3	12
37	Аудио сопровождение игры. Фоновая музыка. Звуки окружения.	3	1	2
38	Основы освещения и постановки света и камеры. Skybox.	3	2	1
39	Animator Blend Tree. Смешивание анимаций.	6	3	3
40	Кат-сцены. Timeline.	6	2	4
41	Постобработка. Post processing.	3	1	2
42	Реализация настроек игры.	6	3	3
43	Публикация проекта.	3	1	2
Итоговое тестирование (3 часа)				
44	Подготовка к защите работы	1	0	1
45	Защита работы, итоговое тестирование	2	0	2
Итого часов:		144	69	75

СОДЕРЖАНИЕ ПРОГРАММЫ

1. Командная работа

1.1. Введение. Знакомство. Образование команд.

Знакомство учеников друг с другом. Образование команд для дальнейшей работы над проектами.

1.2. Soft skills. Работа в команде. Командообразование. Team Leader.

Что такое Soft skills? Основы командной работы и командообразования. Кто такой team leader и какие у него обязанности.

1.3. Основы проектной деятельности. Основы систем ведения проекта Kanban, Scrum и Agile.

Проектная работа, основные особенности работы в команде в IT. Сравнение и разбор основных систем ведения проектов: Kanban, Scrum, Agile.

1.4. Системы контроля версий. Git.

Что такое системы контроля версий. Основы работы с Git. Ветки и слияния. Пуш проекта в удалённый репозиторий.

2. Работа над окружением и наполнением игры

2.1. Введение в основы геймдизайна. Различные игровые механики на примерах из игр.

Что такое геймдизайн? Основы геймдизайна на примерах из игр.

2.2. Основы сценаристики и нарративного дизайна.

Разбор и сравнение основных приёмов, шаблонов и терминов сценаристики и нарративного дизайна.

2.3. Основы левелдизайна. Создание запоминающейся локации.

Что такое левелдизайн? Паттерны и шаблоны для создания интересных и запоминающихся локаций.

2.4. Путь игрока. Приёмы и паттерны. Сторителлинг через окружение.

Из чего состоит локация. Как рассказать историю через окружение. Путь игрока и триггеры для игрока.

2.5. Прототипирование уровней. ProBuilder.

Основы работы с ProBuilder для создания прототипа уровня.

2.6. Terrain. Работа с ландшафтом.

Что такое Terrain. Основные инструменты для работы с ландшафтом. Создание растительности.

2.7. Terrain System. Инструменты. Карты высот.

Модернизация системы Terrain. Использование новых инструментов. Ипортирование карты высот и создание на их основе ландшафта.

3. Продвинутое программирование и моделирование

3.1. Основы ООП. 4 базовых принципа ООП. Принцип 1. Абстракция. Художественные стили моделей. Стилизация и реализм.

Что такое ООП? 4 базовых принципа объектно-ориентированного программирования. Что такое абстракция? Виды художественных стилей. Особенности реализма и стилизации моделей.

3.2. Принцип 2. Инкапсуляция. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.

Что такое инкапсуляция? Реализация сокрытия полей с предоставлением доступа. Различные способы текстурирования моделей. Особенности текстурирования в разных стилях. Особенности работы в софте для текстурирования: Quixel Mixer, Substance Painter, Blender.

3.3. Свойства в C#. Инкапсуляция через свойства. Модификаторы get и set. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.

Что такое свойства? Применение свойств для реализации инкапсуляции. Модификаторы на чтение (get) и запись (set). Различные способы текстурирования моделей. Особенности текстурирования в разных стилях. Особенности работы в софте для текстурирования: Quixel Mixer, Substance Painter, Blender.

3.4. Принцип 3. Наследование. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.

Что такое наследование? Переиспользование кода. Архитектура кодовой базы с использованием наследования. Различные способы текстурирования моделей. Особенности текстурирования в разных стилях. Особенности работы в софте для текстурирования: Quixel Mixer, Substance Painter, Blender.

3.5. Абстрактные классы. Абстрактные и виртуальные методы. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.

Что такое абстрактный класс? Отличие абстрактного класса от обычного. Реализация и применение абстрактных и виртуальных методов. Переопределение. Различные способы текстурирования моделей. Особенности текстурирования в разных стилях. Особенности работы в софте для текстурирования: Quixel Mixer, Substance Painter, Blender.

3.6. Интерфейсы в C#. Абстракция vs Наследование. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.

Что такое интерфейсы? Применение интерфейсов для реализации композиции. Сравнение подходов реализации геймлея с помощью Композиции и Наследования. Различные способы текстурирования моделей. Особенности текстурирования в разных стилях. Особенности работы в софте для текстурирования: Quixel Mixer, Substance Painter, Blender.

3.7. Обычные классы vs MonoBehaviour. Сравнение и примеры использования. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.

Использование классических классов (без наследования MonoBehaviour). Сравнение подходов. Разбор на примерах геймплейных механик. Различные способы текстурирования моделей. Особенности

текстурирования в разных стилях. Особенности работы в софте для текстурирования: Quixel Mixer, Substance Painter, Blender.

3.8. Принцип 4. Полиморфизм. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.

Что такое полиморфизм? Переопределение типов. Принцип подстановки Барбары Лисков. Различные способы текстурирования моделей. Особенности текстурирования в разных стилях. Особенности работы в софте для текстурирования: Quixel Mixer, Substance Painter, Blender.

3.9. Принципы для разработки: KISS, DRY, YAGNI, BDUF, SOLID, APO. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.

Обзор основных принципов и подходов к разработке проекта: KISS, DRY, YAGNI, BDUF, SOLID, APO. Различные способы текстурирования моделей. Особенности текстурирования в разных стилях. Особенности работы в софте для текстурирования: Quixel Mixer, Substance Painter, Blender.

3.10. События в C#. Реализация взаимодействия между скриптами. Способы текстурирования. Детальное, процедурное, текстурные атласы. Работа с Quixel Mixer, Substance Painter, Blender.

Что такое делегаты и события? Вызов событий. Подписка на событие. System.Action. Различные способы текстурирования моделей. Особенности текстурирования в разных стилях. Особенности работы в софте для текстурирования: Quixel Mixer, Substance Painter, Blender.

3.11. Работа с коллекциями. Массивы vs Списки. Дополнительные структуры хранения данных. Создание UDIM развёртки.

Сравнение массивов и списков. В каких ситуациях что лучше использовать? Обзор дополнительных структур хранения: очередь, стек, словарь. Что такое UDIM развёртка и для чего она нужна? Создание UDIM развёртки на примере модели.

3.12. Работа с файлами в C#. Чтение и запись файлов. Работа с файлами в Unity. Риггинг, скиннинг и анимация.

Изучение принципов работы с файлами из кода на языке C#. StreamReader и StreamWriter. Класс File. Что такое риггинг и скиннинг? Особенности риггинга и скиннинга для различных художественных стилей моделей. Основы анимации модели.

3.13. Scriptable Objects. Применение в архитектуре игры. Риггинг, скиннинг и анимация.

Что такое Scriptable Object? Создание шаблона. Использование Scriptable Object в архитектуре игры. Что такое риггинг и скиннинг? Особенности риггинга и скиннинга для различных художественных стилей моделей. Основы анимации модели.

4. Паттерны проектирования, продвинутое моделирование

4.1. Паттерны проектирования. Паттерны «Абстрактная фабрика» и «Фабричный метод». Риггинг, скиннинг и анимация.

Что такое паттерны? Применение паттернов для упрощения понимания кодовой базы. Виды паттернов. Разбор и реализация паттернов «Абстрактная

фабрика» и «Фабричный метод» на примере реализации игровых механик. Что такое риггинг и скиннинг? Особенности риггинга и скиннинга для различных художественных стилей моделей. Основы анимации модели.

4.2. Паттерн «Машина состояний». 3D сканирование. Оптимизация сканов и моделей для игр.

Разбор и реализация паттерна «Машина состояний» на примере реализации игровых механик. 3D сканирование в различных программах. Оптимизация сетки полученных моделей и дальнейшая доработка для использования в видеоиграх.

4.3. Паттерн «Наблюдатель». Постановка света, камеры и окружения.

Разбор и реализация паттерна «Наблюдатель» на примере реализации игровых механик. Основы постановки освещения в кадре. Работа с камерой и расположением сцены в кадре.

4.4. Паттерн «Медиатор». Постановка света, камеры и окружения.

Разбор и реализация паттерна «Медиатор» на примере реализации игровых механик. Основы постановки освещения в кадре. Работа с камерой и расположением сцены в кадре.

5. Продвинутая работа с движком, продвинутое моделирование

5.1. Архитектура проекта. Иерархия и структура файловой системы. Постановка света, камеры и окружения.

Хранение файлов в проекте. Файловая структура проекта. Разделение скачиваемых плагинов и файлов проекта. Основы постановки освещения в кадре. Работа с камерой и расположением сцены в кадре.

5.2. Создание пользовательских пакетов и инструментов для движка. Создание рендера, анимированного превью и тизера.

Unity Editor. Работа с Gizmos. Создание инструментов, ускоряющих рутинную работу в движке. Создание пользовательских пакетов и их импорт в проект. Что такое рендер, тизер и анимированное превью? Основы создания тизера. Создание тизера или анимированного превью на примере созданной сцены.

5.3. Input System. Обработка ввода для разных устройств. Создание рендера, анимированного превью и тизера.

Новая система ввода в Unity. Разделение ввода в Input Map. Что такое рендер, тизер и анимированное превью? Основы создания тизера. Создание тизера или анимированного превью на примере созданной сцены.

5.4. Dotween. Работа с анимациями. Создание рендера, анимированного превью и тизера.

Изучение пакета Dotween. Упрощение работы с анимацией. Что такое рендер, тизер и анимированное превью? Основы создания тизера. Создание тизера или анимированного превью на примере созданной сцены.

5.5. Шейдеры. Shader Graph. Создание кастомных шейдеров. Работа с моделями в движке.

Что такое шейдеры? Работа в Shader Graph. Создание пользовательских шейдеров для игры. Основы работы в движке. Работа над

окружением в игре. Постановка сцены в движении. Настройка материалов на модели.

5.6. Light Probes. Углублённая работа с освещением. Создание и оформление портфолио для 3D художника.

Что такое Light Probes? Отражение света. Разработка проработанного освещения для проекта. Основы оформления «продающего» портфолио для 3D художника.

5.7. Использование LOD. Настройка детализации моделей в проекте. Площадки для публикации. Создание страницы и выгрузка работ.

Что такое LOD? Применение LOD для оптимизации проекта. Настройка детализации моделей в зависимости от расстояния. Обзор различных площадок для публикации 3D моделей. Создание страницы-портфолио на популярных площадках. Основы публикации работ.

6. Проектная деятельность

6.1. Разработка основных геймплейных механик и скелета игры.

Разработка выпускного проекта. Создание всех геймплейных механик. Developer Build.

6.2. Аудио сопровождение игры. Фоновая музыка. Звуки окружения.

Работа с аудио файлами в Unity. AudioManager. Создание звуков во время игры.

6.3. Основы освещения и постановки света и камеры. Skybox.

Настройка освещения для проекта. Realtime и запекаемое освещение. Настройка Skybox. Работа с камерой. Cinemachine.

6.4. Animator Blend Tree. Смешивание анимаций.

Что такое Blend Tree? Реализация плавного перехода между анимациями. Смешивание анимаций.

6.5. Кат-сцены. Timeline.

Что такое кат-сцена? Изучение функционала пакета Timeline. Создание кат-сцен для игры.

6.6. Постобработка. Post processing.

Изучение пакета PostProcessing. Добавление эффектов. Улучшение визуальной составляющей игры.

6.7. Реализация настроек игры.

Создание рабочих настроек для игры: настройки графики, звука, управления.

6.8. Публикация проекта.

Создание билда. Работа с Yandex.Games API. Обзор популярных площадок для публикации проекта. Публикация проекта.

7. Итоговое тестирование

Подготовка к итоговому тестированию. Прохождение теста. Сбор обратной связи.